Configure DCPMM in VMware ESXi for AppDirect Mode

Contents

Introduction

This document describes the process to configure ESXi on Unified Computing System (UCS) B series servers using Intel® Optane™ Persistent Memory (PMEM) in host managed mode.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- UCS B series
- Intel® Optane™ Data Center Persistent Memory Module (DCPMM) concepts
- VMware ESXi and vCenter Server administration

Ensure that you meet these requirements before you attempt this configuration:

- Refer to the PMEM guidelines on the B200/B480 M5 specification guide.
- Ensure the CPU is second generation Intel® Xeon® Scalable processors.
- PMEM/Dynamic Random Access Memory (DRAM) ratio meets requirements as per KB 67645.
- ESXi is 6.7 U2 + Express Patch 10 (ESXi670-201906002) or later. Earlier 6.7 releases are not supported.
- UCS Manager and Server are in a 4.0(4) version or above. For the latest recommended version please visit www.software.cisco.com/.

Components Used

The information in this document is based on these software and hardware versions:

- UCS B480 M5
- UCS Manager 4.1(2b)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

In UCS Servers configured for App Direct mode, VMware ESXi virtual machines access to Optane DCPMM Persistent memories Non-Volatile Dual In-Line Memory Modules (NVDIMMs).

Intel Optane DCPMM can be configured through the IPMCTL management utility through the Unified Extensible Firmware Interface (UEFI) shell or through the OS Utilities. This tool is designed to perform some of the next actions:

- Discover and Manage Modules
- Update and configure Module firmware
- Monitor Health
- Provision and configure Goal, Region, and Namespaces
- Debug and troubleshoot PMEM

UCS can be configured using a persistent memory policy attached to the service profile for ease of use.

The open-source Non-Volatile Device Control (NDCTL) utility is used to manage the LIBNVDIMM Linux Kernel subsystem. The NDCTL utility allows a system to provision and performs configurations as regions and namespaces for OS use.

Persistent memory added to an ESXi host is detected by the host, formatted, and mounted as a local PMem datastore. In order to use the PMEM, ESXi uses the Virtual Machine Flying System (VMFS)-L file system format, and only one local PMEM datastore per host is supported.

Different from other datastores, the PMEM datastore does not support tasks as traditional datastores. The VM home directory with the vmx and vmware.log files cannot be placed on the PMEM datastore.

PMEM can be presented to a VM in two different modes: Direct-Access Mode and Virtual Disk mode.

- Direct-Access Mode
  VMs can be configured for this mode by presenting the PMEMregion in the form of an NVDIMM. VM Operating System must be PMem-aware to use this mode. Data stored on NVDIMM modules can persist across power cycles since the NVDIMM act as byte-addressable memory. NVDIMMs are automatically stored on the PMem datastore created by the ESXi when formating the PMEM.

- Virtual Disk Mode
  Intended for traditional and legacy OS residing on VM in order to support any hardware versions. VM OS is not required to be PMEM-aware. In this mode, a traditional Small Computer System Interface (SCSI) virtual disk can be created and used by the VM OS.

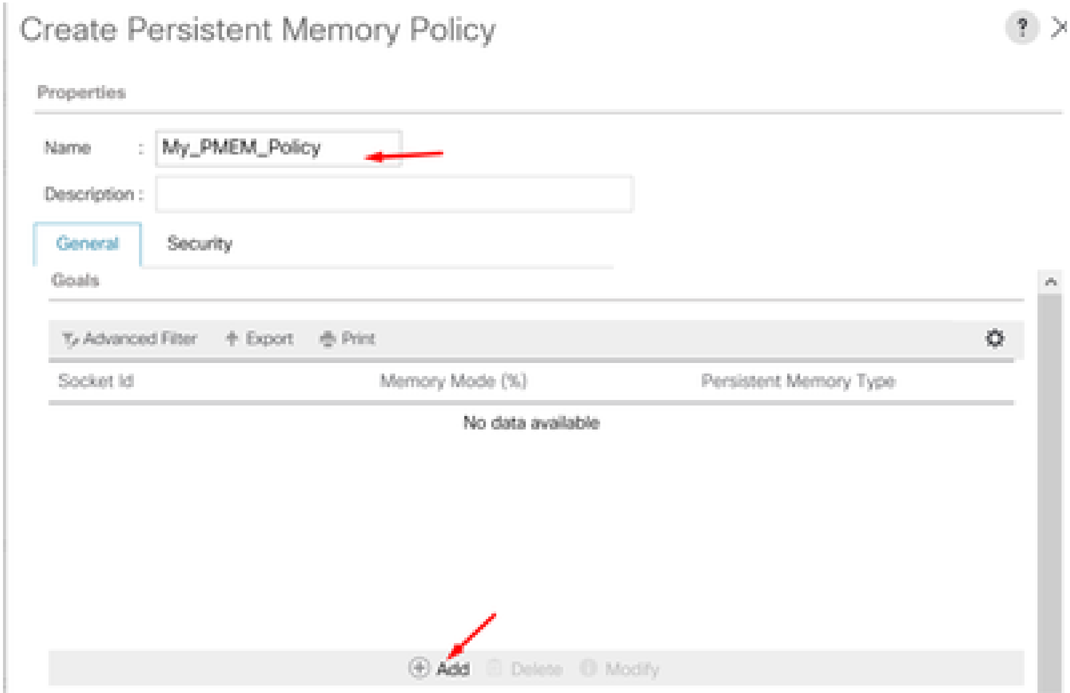
This document describes the configuration to use a Virtual Machine in Direct-Access mode.

Configure

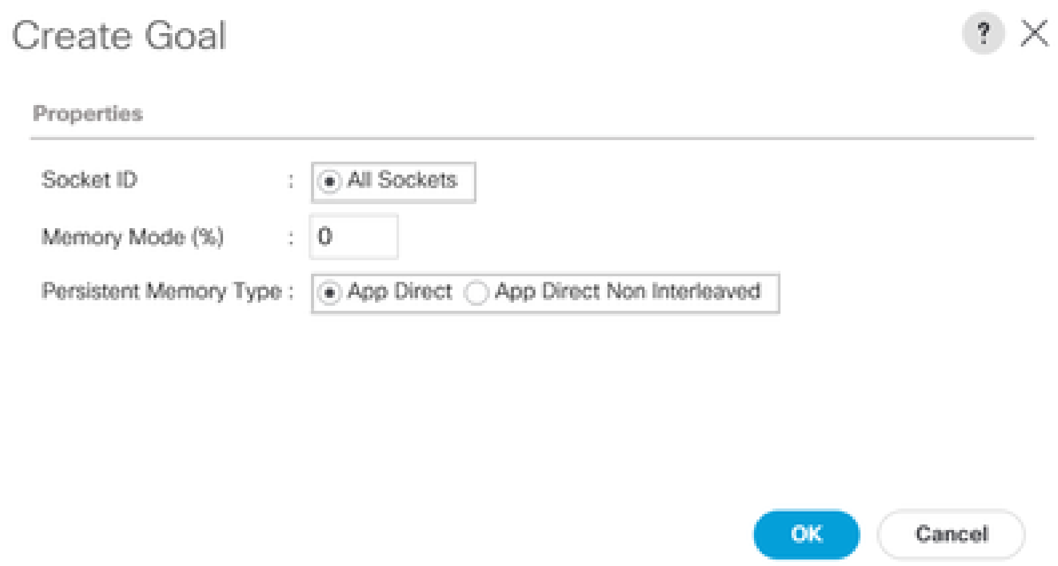
This procedure describes how to configure ESXi on UCS Blade series servers using Intel Optane DCPMM.

Configure the Service Profile

1. In UCS Manager GUI, navigate to **Servers > Persistent Memory Policy** and click on **Add** as shown in the image.



2. Create **Goal**, ensure the **Memory Mode** is 0% as shown in the image.

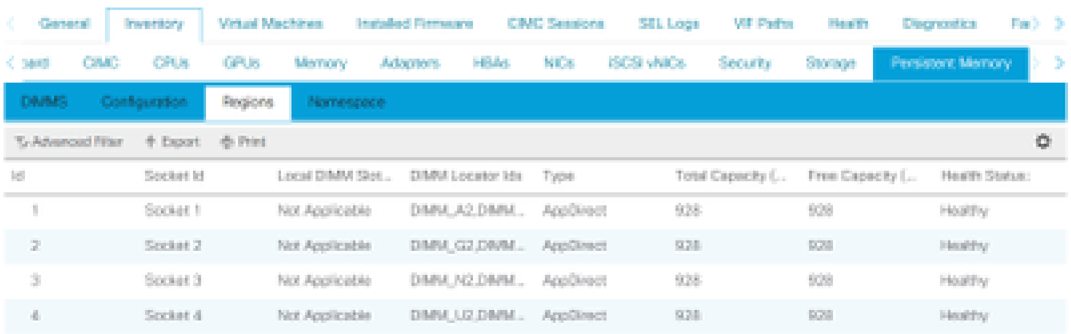


3. Add the PMEM Policy to the desired Service Profile.

Navigate to **Service Profile > Policies > Persistent Memory Policy** and attach the policy created.

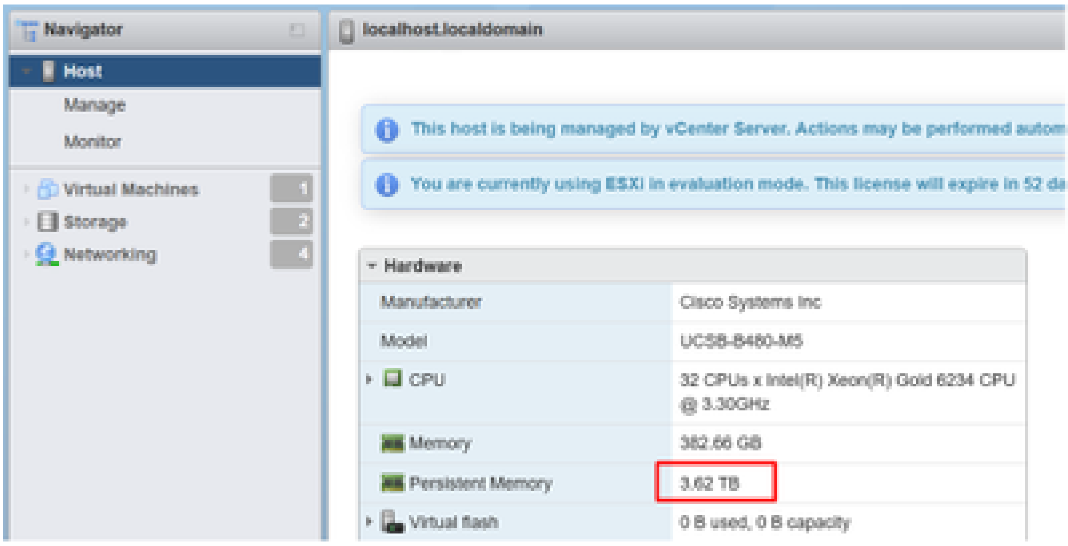
4. Verify the region's health.

Navigate to the selected **Server > Inventory > Persistent Memory > Regions**. The type AppDirect is visible. This method creates one region per CPU Socket.

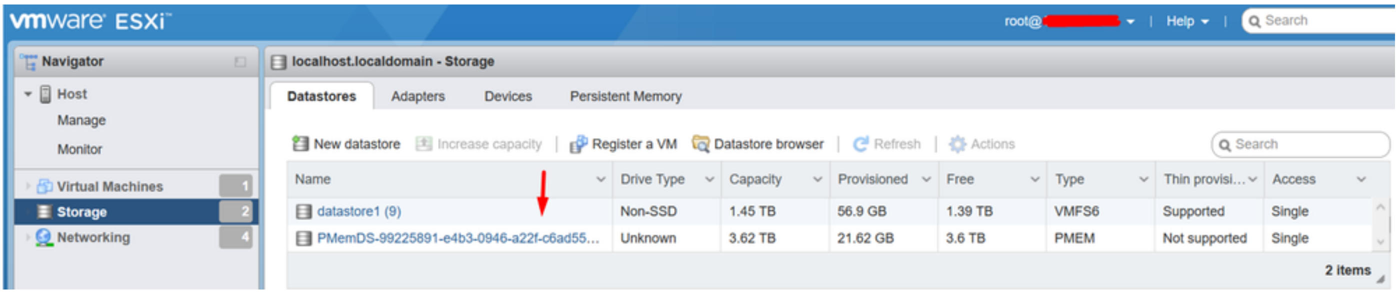


Verify ESXi

1. In the Web console, the host displays the total PMEM available.



2. ESXi displays a special datastore composed of the total amount of PMEM, as shown in the image.

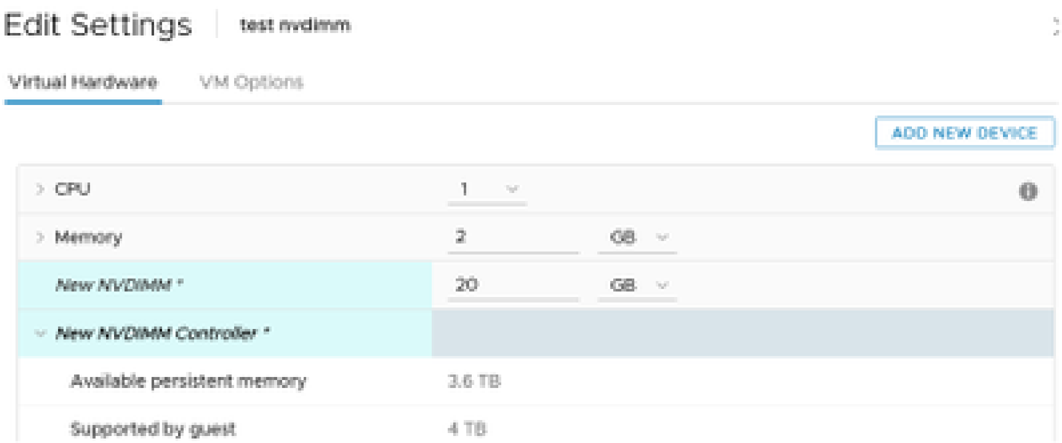


Configure Virtual Machine NVDIMM

1. In ESXi, virtual machines access to Optane DCPMM PMEM as NVDIMMs. In order to assign an NVMDIMM to a virtual machine, access the virtual machine through vCenter and navigate to **Actions > Edit Settings**, click on **ADD NEW DEVICE** and select **NVDIMM** as shown in the image.

Note: When you create a virtual machine, ensure that the OS compatibility meets the minimum required version that supports Intel® Optane™ Persistent Memory, otherwise the **NVDIMM** option does not appear in the selectable items.

2. Set the size of NVDIMM as shown in the image.



Configure Namespace in the Virtual Machine

1. The **NDCTL** utility is used to manage and configure the PMEM or NVDIMM.

In the example, Red Hat 8 is used for configuration. Microsoft has PowerShell cmdlets for persistent memory namespace management.

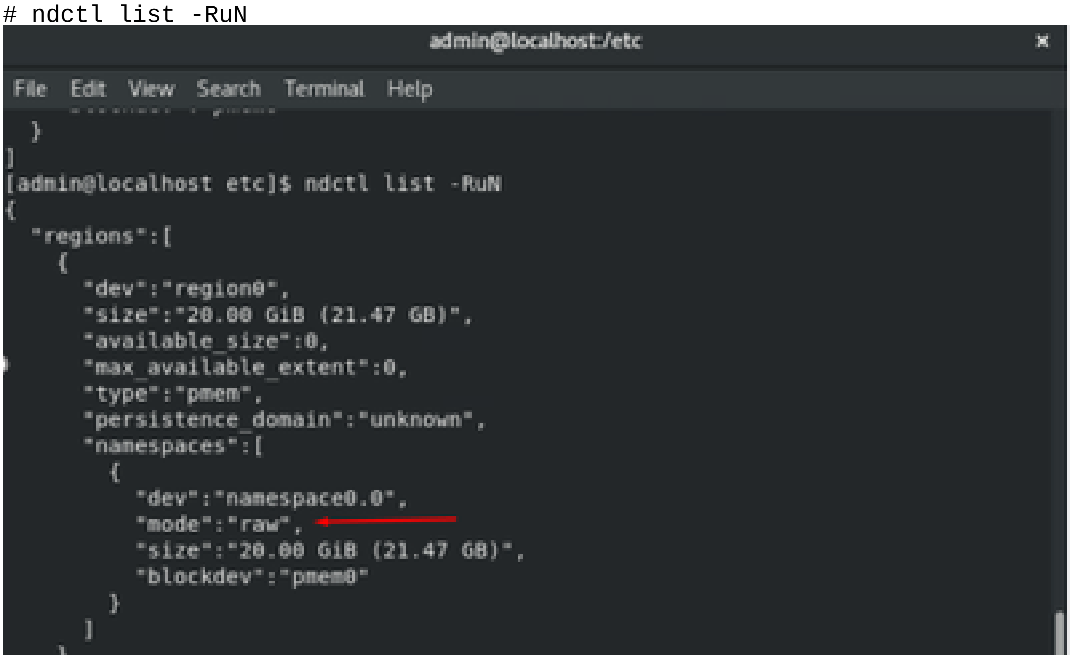
Download the **NDCTL** utility by using the available tool as per the Linux Distribution

For example:

```
# yum install ndctl # zypper install ndctl # apt-get install ndctl
```

2. Verify the NVDIMM region and namespace created by default by ESXi, when the NVDIMM is assigned to the virtual machine, verify space matches with configuration. Ensure the mode of the namespace is set to **raw** this means ESXi has created the namespace. In order to verify, use the command:

```
# ndctl list -RuN
```



3. (Optional) If the namespace has not been created already, a namespace can be created with the command:

```
# ndctl create-namespace
```

The **ndctl create-namespace** command creates a new namespace in **fsdax** mode by default and creates a new **/dev/pmem([x].[y])** device. If a namespace has been created already, this step can be skipped.

4. Select the PMEM access mode, the modes available for configuration are:

- Sector Mode:

  Presents storage as a fast block device, this is useful for legacy applications that are still not able to use persistent memory.

- Fsdax Mode:

  Allows the persistent memory devices to support direct access to the NVDIMM. File system direct access requires the use of **fsdax** mode, in order to enable the use of the direct access programming model. This mode allows the creation of a file system on top of the NVDIMM.

- Devdax Mode:

  Provides raw access to persistent memory using a DAX character device. File systems cannot be created on devices using **devdax** mode.

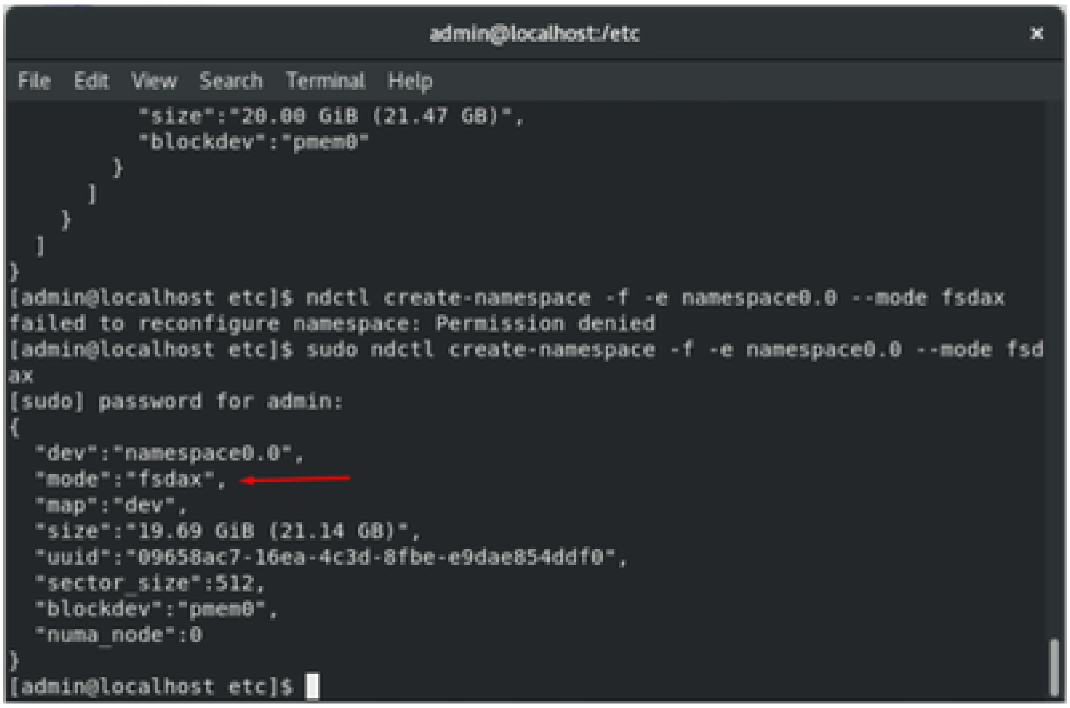
- Raw Mode:

  This mode has several limitations and is not recommended for using Persistent Memory.

In order to change the mode to **fsdax** mode, use the command:

```
ndctl create-namespace -f -e <namespacex.y> --mode fsdax
```

If there is a **dev** already created, the dev namespace is used to format and modify the mode to **fsdax**.



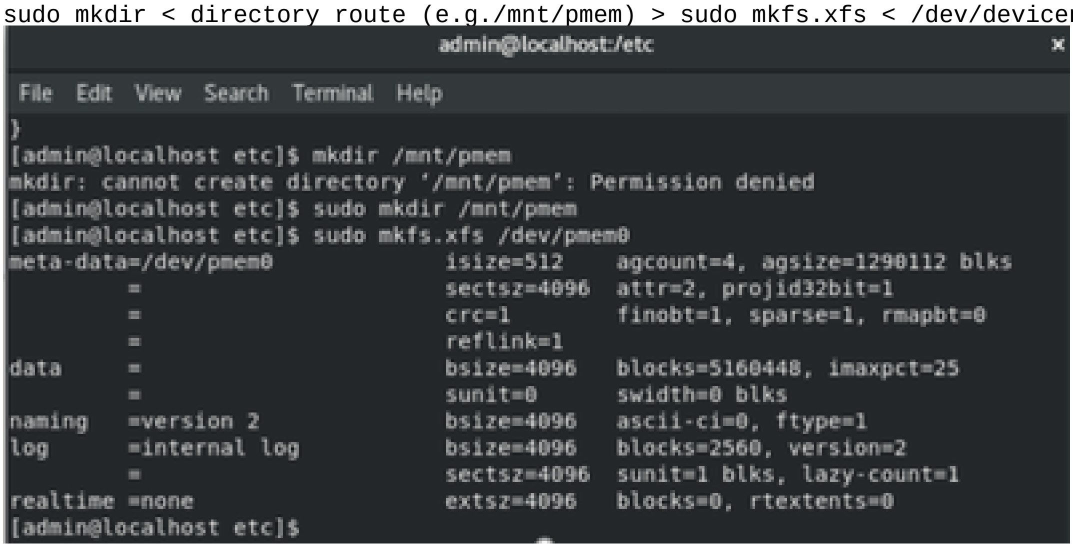
Note: These commands require that the account has root privileges, **sudo** command might be required.

5. Create a directory and filesystem.

Direct Access or DAX is a mechanism that allows applications to directly access persistent media from the CPU (through loads and stores), bypassing the traditional I/O stack. DAX-enabled persistent memory file systems include ext4, XFS, and Windows NTFS.

Example of XFS file system created and mounted:

```
sudo mkdir < directory route (e.g./mnt/pmem) > sudo mkfs.xfs < /dev/devicename (e.g. pmem0) >
```



6. Mount the file system and verify that is successful.

```
sudo mount <file system > < directory > df -h < directory >
```

```
admin@localhost:/etc
File  Edit  View  Search  Terminal  Help
[admin@localhost etc]$ sudo mount /dev/pmem0 /mnt/pmem/
[admin@localhost etc]$ // verify the mount was successful
bash: //: Is a directory
[admin@localhost etc]$ df -h /mnt/pmem/
Filesystem      Size  Used Avail Use% Mounted on
/dev/pmem0       20G  173M   20G   1% /mnt/pmem
[admin@localhost etc]$
```

The VM is ready to use PMEM.

Troubleshoot

Is generally recommended to mount this DAX-enabled file system using the **-o dax** mount option, if an error is found.

```
[admin@localhost etc]$ sudo mount -o dax /dev/pmem0 /mnt/pmem/
mount: /mnt/pmem: wrong fs type, bad option, bad superblock on /dev/pmem0, missi
ng codepage or helper program, or other error.
```

Filesystem repair is executed to ensure integrity.

```
[admin@localhost etc]$ sudo xfs_repair /dev/pmem0
[sudo] password for admin:
Phase 1 - find and verify superblock...
Phase 2 - using internal log
        - zero log...
        - scan filesystem freespace and inode maps...
        - found root inode chunk
Phase 3 - for each AG...
        - scan and clear agi unlinked lists...
        - process known inodes and perform inode discovery...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
        - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
        - setting up duplicate extent list...
        - check for inodes claiming duplicate blocks...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
Phase 5 - rebuild AG headers and trees...
        - reset superblock...
Phase 6 - check inode connectivity...
        - resetting contents of realtime bitmap and summary inodes
        - traversing filesystem ...
        - traversal finished ...
        - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
[admin@localhost etc]$
```

As a workaround, the mount can be mounted without the **-o dax** option.

> **Note**: In **xfsprogs** version 5.1, the default is to create XFS file systems with the **reflink** option enabled. Previously it was disabled by default. The **reflink** and **dax** options are mutually exclusive which causes the mount to fail.

"DAX and reflink cannot be used together!" the error can be seen in **dmesg** when the mount command fails:

admin@localhost:/etc

File Edit View Search Terminal Help

```
log      =internal log          bsize=4096   blocks=2560, version=2
         =                      sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
[admin@localhost etc]$ mount -o dax /dev/pmem0 /mnt/pmem
mount: only root can use "--options" option
[admin@localhost etc]$ sudo mount -o dax /dev/pmem0 /mnt/pmem/
mount: /mnt/pmem: wrong fs type, bad option, bad superblock on /dev/pmem0, missi
ng codepage or helper program, or other error.
[admin@localhost etc]$ dmesg -T | tail
[mar nov 10 00:12:18 2020] VFS: busy inodes on changed media or resized disk sr0
[mar nov 10 00:12:22 2020] ISO 9660 Extensions: Microsoft Joliet Level 3
[mar nov 10 00:12:22 2020] ISO 9660 Extensions: RRIP_1991A
[mar nov 10 01:47:35 2020] pmem0: detected capacity change from 0 to 21137195008
[mar nov 10 01:51:19 2020] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use
at your own risk
[mar nov 10 01:51:19 2020] XFS (pmem0): DAX and reflink cannot be used together!
[mar nov 10 01:53:06 2020] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use
at your own risk
[mar nov 10 01:53:06 2020] XFS (pmem0): DAX and reflink cannot be used together!
[mar nov 10 01:59:29 2020] XFS (pmem0): DAX enabled. Warning: EXPERIMENTAL, use
at your own risk
[mar nov 10 01:59:29 2020] XFS (pmem0): DAX and reflink cannot be used together!
[admin@localhost etc]$
```

As a workaround, remove the **-o dax** option.

admin@localhost:/etc

File Edit View Search Terminal Help

```
[admin@localhost etc]$ sudo mount /dev/pmem0 /mnt/pmem/
[admin@localhost etc]$ // verify the mount was successful
bash: //: Is a directory
[admin@localhost etc]$ df -h /mnt/pmem/
Filesystem      Size  Used Avail Use% Mounted on
/dev/pmem0       20G  173M   20G   1% /mnt/pmem
[admin@localhost etc]$
```

Mount with ext4 FS.

The EXT4 file system can be used as an alternative because it does not implement the reflink feature but support DAX.

```
[admin@localhost etc]$ sudo mkfs.ext4 /dev/pmem0
mke2fs 1.44.3 (10-July-2018)
/dev/pmem0 contains a xfs file system
Proceed anyway? (y,N) y
Creating filesystem with 5160448 4k blocks and 1291808 inodes
Filesystem UUID: 164c6d57-0462-45a0-9b94-703719272816
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[admin@localhost etc]$ sudo mount /dev/pmem0 /mnt/pmem/
[admin@localhost etc]$ df -h /mnt/pmem/
Filesystem      Size  Used Avail Use% Mounted on
/dev/pmem0       20G   45M   19G   1% /mnt/pmem
[admin@localhost etc]$
```

Related Information

- **Quick Start Guide: Provision Intel® Optane™ DC Persistent Memory**
- **Persistent Memory configuration**
- **Management Utilities ipmctl and ndctl for Intel® Optane™ Persistent Memory**
- **Technical Support & Documentation - Cisco Systems**